A Message Interchange Protocol based on Routing Information Protocol in a Virtual World

Jui-Fa Chen+, Wei-Chuan Lin*, Hua-Sheng Bai+, Shih-Yao Dai+
+Department of Information Engineering, TamKang University
+E-mail: alpha@mail.tku.edu.tw
*Department of Information Technology, Tak-Ming College,
*E-mail: wayne@takming.edu.tw

Abstract

*As the network transmission is prevailing in many applications, an entity in a network virtual environment (**NVE**) can deliver data to other entities under peer-to-peer architecture. This paper proposes a message interchange protocol based on routing information protocol (**RIP**). In this message interchange protocol, every entity can know that it is in some entities' area of interested (**AOI**). In this way, every entity can deliver data to other entities without via server. Furthermore, according to the group in four quadrants, every entity can get the information of other entities' by interchanging data of each others'. Based on this message interchange protocol, every entity can still work in the NVE when the server is crashed. Even if there is an entity crashes, other entities can also work in the NVE.*

Keyword: message interchange protocol, routing information protocol, area of interest, group in four quadrants

1. Introduction

Network virtual environment (NVE) is very popular among science fiction fans and it has been pictured out in many books and movies in which is known as "The Matrix" [Wachowski 03]. Most of the NVE adopts the centralized Client-Server architecture, such as [Huang 03][Faisstnauer 00][Funkhouser 96]. In a centralized Client-Server Architecture, when an entity wants to deliver data to other entities, it first delivers data to a server, and the server delivers the data to other entities. In this method, the server should know who are interested in the data of this entity's. It means that the server should calculate the AOI of all clients. However, it takes a long time for the server to calculate the AOI of all clients and this is a heavy load for the server. Moreover, because entities should deliver data to other entities via the server, the network bandwidth of server is heavy [Signhal 97] [Signhal 99]. In additional, because all of the data delivered is by the server, if server is failed, the system is crashed.

In order to reduce the network bandwidth and the computation load of the server in the Client-Server architecture, the entities should deliver data to other entities by the peer-to-peer architecture. Therefore, this paper proposes a message interchange protocol based on routing information protocol (RIP) to interchange data of the entities by peer-to-peer architecture.

In this paper, section 2 is the related works. Section 3 is the proposed system architecture. Section 4 is the simulation results of the proposed peer-to-peer architecture. Finally, the section 5 is the conclusions and future works.

2. Related Works

In this section, the related works are discussed such as Area of Interested (AOI), Routing Information Protocol (RIP) and Connected Graph. The Peer-to-Peer architecture is also discussed.

2.1 Area of Interested

In client-server architecture, the client can deliver data to other clients via server. The numbers of user are restricted by the performance and bandwidth of the server. AOI (Area of Interested) is used as a filtering mechanism [Macedonia 95] to avoid sending data to unrelated entities. By the AOI mechanism, the entity does not have to deliver data to the entities outside AOI. In this way, it can promote the performance of the server.

2.2 Routing Information Protocol

Routing information protocol is a network routing protocol base on the Bellman-Ford (or distance vector) algorithm [Malkin 98] [Hawkinson 96]. According to the RIP, a router will recode the connected routers in its routing table. When the destination of the packet is not in routing table, the router will deliver it to neighbor routers. In order to reduce the transmission cost, the router would periodic interchange routing table with other routers. The proposed mechanism is based on the concept of RIP to reduce the transmission

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

overhead and request of server. The proposed mechanism is to let every entity maintain its own tables. These entities would interchange tables with other entities periodically. In this way, entities can deliver data to other entities without via server.

2.3 Connected Graph and Topology

Let G = (V, E) be an undirected graph. The G is connected if there is a path between any two distinct vertices of G [Grimaldi 99]. In the proposed architecture, V can be a set of entities, E be a set of connectivity between entities and G = (V, E) is a topology in a virtual world. If the topology is connected, every entity can get the information of other entities by connected entities.

2.4 Toward a Peer-to-Peer Shared Virtual Reality

In 2002, Keller proposed a method to build a virtual world [Keller 02]. The concept of this paper is that if an entity detects that entity A enters the AOI of another entity B's, the entity would send the information of entity B to A. In this way, entity A can know that it needs to send its data to entity B. However, if an entity exists a section that there is no entity that can detect it, it would hardly know that an entity is arrived in this section.

The author also proposed a method of Global Connectivity to solve this problem. The method is to make each entity should keep the last one entity in each 180 degree. In this way, it can detect entities come from all direction. If an entity cannot keep the last one entity, it will ask the nearest entities to find an entity can satisfy the Global Connectivity as shown in Fig. 1.

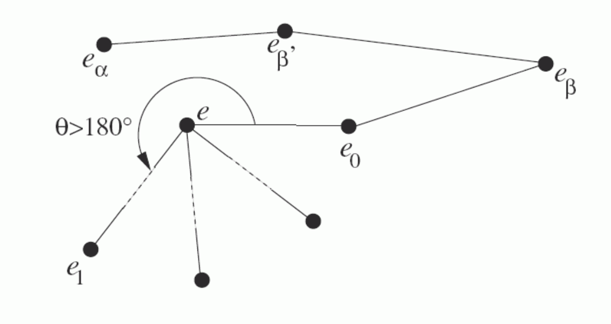


Fig. 1 Global connectivity

3. System Architecture

In order to reduce the network bandwidth and computing loading of the server in the client-server architecture, entities should deliver data to other entities by peer-to-peer architecture. By this method, a server does not need to calculate all client's AOI and then deliver data for entities. In this way, this method can reduce the network bandwidth and the computing loading of the server.

In peer-to-peer architecture, the server should compute the AOIs for entities. How an entity knows which entities it should deliver data to? To solve the problem on Peer-to-Peer architecture, this paper proposes a message interchange protocol based on routing information protocol. By this mechanism, every entity can know that it is in some entities' AOI. Every entity can deliver data to other entities without via server. In additional, if server is failed, the system can still work.

3.1 Virtual World Architecture

In this paper, the whole virtual world is built by many regions [Chen 04]. Every region is controlled by one server. The server records all data in the region. It includes the user's information and the object's data in the virtual world. In this architecture, the server is not actively to obtain the entities' data. It just passively receives data from entities and delivers data to the new entities when they are joined to this region.

3.2 Message Interchange Protocol

In this paper, the discussion is focus on an entity how to deliver data to other entities in the same region. In the proposed message interchange protocol, every entity has three tables to record the information of other entities. These tables are Observable Table (OT), Global Table (GT), and Group Leader Table (GLT). The OT records the entities that can see me. The entity is only deliver data to the entities which was in the OT. The GT records some entities' information that is not immediately needed in OT. The entity can immediately extract the needed entities information from Global Table when the entities' information is needed in OT. Suppose there is a entity B who in the entity A's GT, when entity B is interested in entity A, entity B would be move from the GT of entity A to the OT of entity A. On the contrary, if there is a entity C who in the entity A's OT, when C is not interested in A, entity C will be move from the OT of entity A to the GT of entity A. In this way, entities can know that there are new entities that would be near themselves and update their OT and GT tables. The GLT is designed to maintain the topology connectivity in virtual world. Every entity would record the leaders of four groups. The entities are divided according to the position in my four quadrants and the entities in the same quadrant would form a group. For every group, an entity is chosen as a leader who is the nearest entity to maintain the topology connectivity. The GLT is periodically updated to avoid topology disconnection.

3.2.1 Observable Table

When the attributes of entity are changed, the changed attributes should be synchronized with other entities to maintain identical data in the virtual world. Therefore, the entity can deliver data to appropriate entities by referring OT. The definition of OT is listed as followed:

$$\forall e_1 \in OT(e) \Leftrightarrow d(e,e_1) \le \gamma_{e_1} and \quad e_1 \in E \qquad (3.1)$$

e_1 : An entity's code name in the Virtual World, $for \quad i \in N \cup \phi$

E : A set of entity in one area,

d (e, e_1) : The distance between entity e and entity e_1,

γ_{e_1} : An entity's Aura,

$OT(e_1)$: A set of Observable Table.

In Fig. 2, entity e is in the AOI of Entity e1 and e2, therefore e1 and e2 is recorded in the OT of entity e. When the attributes of entity e are changed, entity e would reference its OT and deliver it's attributes to e1 and e2. However, entity e3 is not in the AOI of any entity, the OT of entity e3 is empty.

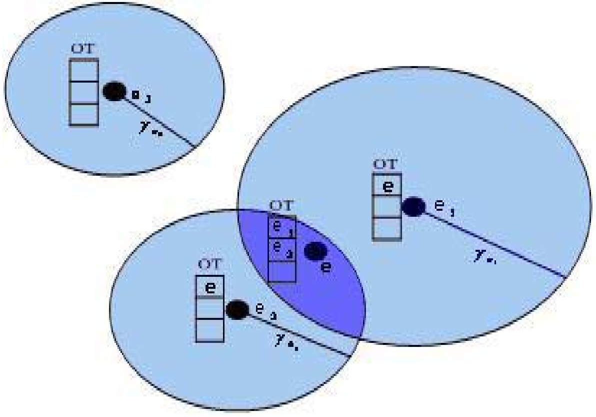


Fig. 2 The example of OT

3.2.1.1 Interchange protocol between entities

As the entity maybe arbitrary moving in the virtual world, it would change its position frequently. An entity may enter the AOI of other entities, or leave the AOI of other entities. Therefore OT maybe incorrect for a while and should be updated at the right moment. For keep the correct of OT, an entity should interchange its OT with other entities' OT for a period of time.

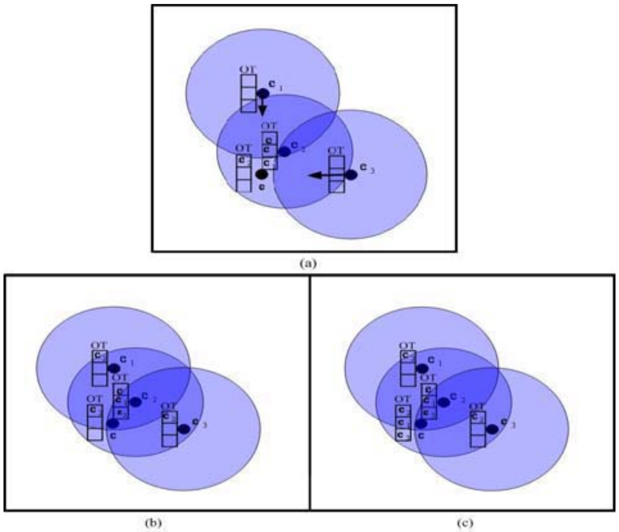


Fig. 3 Interchange OT

If entity e1 is interested in entity e, other entities that got interested in entity e1 maybe also be interested in entity e. In Fig. 3(a), suppose entity e is in the AOI of entity e2, and entity e2 is in the AOI of entity e1 and e3, entity e1 and e3 are moving toward e. In Fig. 3(b), when entity e1 and e3 are close enough to entity e in the virtual world, entity e is in the AOI of entity e1 and e3. When the entity e matches the period time of interchange OT with entity e2, entity e would discover entity e1 and e3 are interested in itself, and entity e record entity e1 and e3 into its OT.

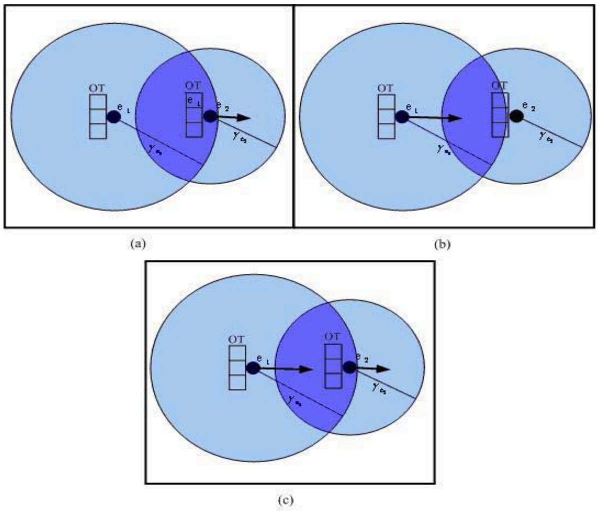
3.2.1.2 The size of Observable Table

Because the entities in the OT are interested in this entity, the entity should keep the connection with them. The entity cannot arbitrarily delete any entity in OT. The entities in the OT can be deleted only when the entities that are not got interested in this entity. The size of OT should be dynamic for this reason. In addition, every entity is occupied a space in virtual world and they cannot overlap. If entities can overlap, they may have infinite entities in same space, and the size of OT may become very large to accommodate these entities in same space.

3.2.2 Global Table

On Peer-to-Peer architecture there are some drawbacks by only using OT. The OT would periodically interchange the information for keeping the correctness of OT. If the interchanging period is too long, the OT would be incorrect. If the interchanging period is too short, the synchronization overhead will occur, and this would raise the entity's load and network traffic. For some case, such as an entity leaves the AOI of some entities, and reenters the

AOI of this entity, this would cause the data lost because the entities connection cannot reconnect soon. In Fig. 4(a), suppose an entity e is in the AOI of entity e1. In Fig. 4(b), entity e is leaves the AOI of entity e1, and the entity e1 would be deleted from the OT of entity e. In Fig. 4(c), if the entity e1 enters the AOI of entity e again, but the interchange period time is not time-up, the entity e cannot know that it is interested by e1. Entity e would not send data to e1, and this may cause the data lost.



(a)

(b)

(c)

Fig. 4 Reenter AOI

To solve this problem, the Global Table (**GT**) is defined. An entity's GT is to store some entities' data that are not immediately needed in OT when this entity interchanges data with other entities. An entity can immediately extract data from GT when the data is needed in OT. The definition of GT is listed as followed:

$$\forall e_1 \in GT(e) \Rightarrow d(e,e_1) > \gamma_{e_1} and\ (e_1 \in E) and (e \in GT(e_1) ore \in OT(e_1)) \tag{3.2}$$

e_1 : An entity's code name in the Virtual World,

$for \quad i \in N \cup \phi$

GT(e_1) : A set of Global Table,

E : A set of entity in one area,

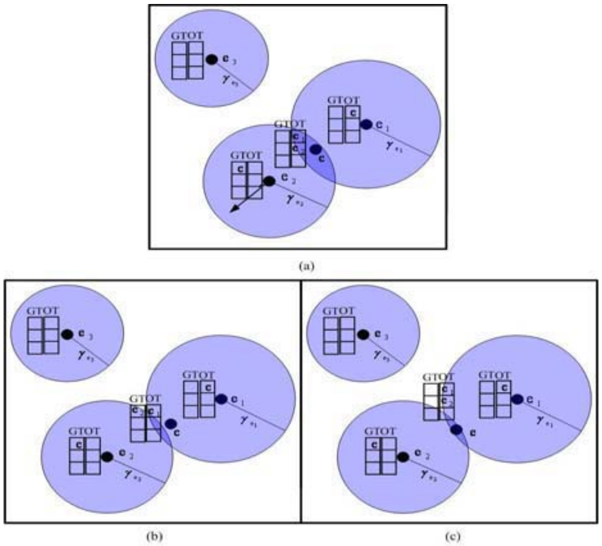
d (e, e_1) : The distance between entity e and entity e_1,

γ_{e_1} : An entity's Aura,

OT(e_1) : A set of Observable Table.

3.2.2.1 Relation between GT and OT

The explanation of the update step of the GT is shown as Fig. 5. In Fig. 5(a), suppose there is an entity e in the AOI of entity e2, and entity e2 in the OT of entity e. In the Fig. 5(b), entity e is left the AOI of entity e2, this cause entity e to move entity e2 from its OT to GT. In the Fig. 5(c), entity e is reentered in the AOI of entity e2, entity e can immediately move entity e2 from its GT to OT and rebuild the network connection.



(a)

(b)

(c)

Fig. 5 Interchange tables between OT and GT

However, only entity e record entity e2 to its GT is not enough. Because when entity e is in the AOI of some entities, these entities may also in the AOI of entity e. When entity e records these entities to its GT, it needs to notify these entities to record entity e to their GT. If only entity e records these entities to its GT, this may cause that entity e cannot get the new data of these entities just because these entities do not know the entity e and they would not send new data to entity e.

3.2.2.2 Interchange protocol between entities

As mentioned before, if the interchange period is too long, the OT would be incorrect. If the interchange period is too short, the synchronization overhead would occur, and this may raise the load of entities and network traffic. Moreover, if the OT of the entity is empty, this entity can not know other entities. To solve these problems, the table interchange methods between entities are listed as followed:

- If OT is not empty, the OT should be interchanged between entities.
- If OT is empty but the GT is not empty, the GT should be interchanged between entities.
- After the table interchanged between entities, this can keep entities in the corresponding tables of entities.

3.2.2.3 Data Update mechanism for the entities

Although the entities in the GT are not the needed information of the entity right now, if the entity is interested in them, the information of them can be obtain from the entity's GT immediately. However, these entities are not stay at same position in the virtual world, they may arbitrarily move. In such circumstances, the information in the GT may not be up-to-date, this may cause the inconsistency between entities in the virtual world. To avoid this problem, the entities in this entity's GT should also need to be updated.

The proposed method is to update the information of entities in the entity's GT periodically. The period is according to the distance between entities. For the near entities of the entity, they need more accurate. .The update frequency should be high. For the far entities of the entity, the update frequency should be low. The Dead Reckoning algorithm can also be used to reduce the frequency of synchronization.

3.2.2.4 The size of Global Table

When an entity interchanges data with other entities as time goes by, its GT maybe full. If the size of GT is not limit, it would contain all the entities in the virtual world and these entities may not near the entity itself. To interchange data with these entities is useless; therefore, the size of GT is limited. When the GT is full, the simple way is to remove the most far away from the current entity. However, this way cause problem such as in Fig. 6. Suppose entity e2 is the most far away from other entities, when other entity's GT is full, they will first remove entity e2 form their GT. Finally, all entities will not know entity e2, and entity e2 become an orphan entity.

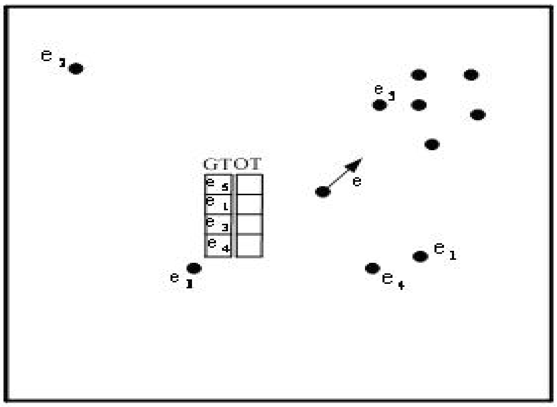


Fig. 6 The orphan entity example

For this reason, the Group in Four Quadrants Algorithm is proposed to drop data in GT. The algorithm is briefly explained as followed:

For the entity in the virtual world, the entity is used as center to divide the entity in the virtual world into four Quadrants, and the entities in the same Quadrant can form a group as shown in Fig. 7. For each group, an entity is chosen as a group leader entity to be record into the GT of the entity. The chosen entity is the nearest to the entity and it is not in the OT of the entity. By the interchange protocol, the entity can get all entities' information.

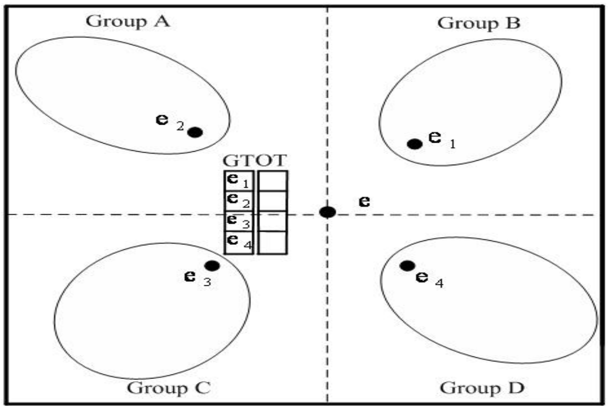


Fig. 7 The Group in Four Quadrants

According to this algorithm, the entity which is the most far away from the entity can be dropped and the dropped entity should not be the only one entity in this quadrant. In this way, it can avoid the orphan entity and keep connection of the virtual world.

3.2.3 Group Leader Table

According to the four quadrants algorithm, every entity recodes each group's leader in its GT. The group leader entity is the nearest entity to the entity and it is not in the OT of the entity. Because of the group leader entity may leave its group, the entity cannot get other entities' information in the same group from the group leader entity. In Fig. 8(a), suppose there is a entity e records four group leader entities e1, e2, e3 and e4. In Fig. 8(b), entity e1 as the group leader in group B moved to group D and become the member of group D. In Fig. 8(c), when the entity e comes to Group B, it would need the other entities' information in the Group B. However, entity e1 is far away from Group B and entity e1 may not have any entities' information in group B. The entity e cannot get any information from entity e1. This causes the data inconsistency between entities.

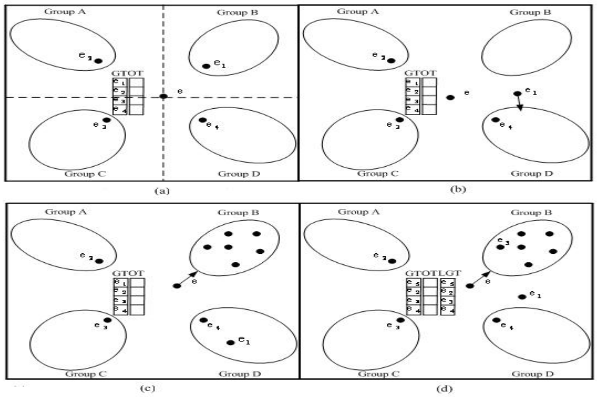


Fig 8 Illustrate of the group leader table

For this reason, the Group Leader Table (**GLT**) is defined to solve this problem. The entity not only store four groups' leader in its GT, but also store them in its GLT. The new group leader's information can be obtained after the original group generation its new leader by the leader election algorithm after a dynamic time quantum. In Fig. 8(d), as time goes by, entity e ask entity e1 who is the new leader of group B. Entity e1 would reply entity e that the new group leader is entity e5, and entity e update its GT. When entity e comes to group B, it can obtain other entities of Group B from entity e5. The definition of GLT is listed as followed:

$$\forall e_1 \in GLT(e) \Rightarrow d(e,e_1) > \gamma_{e_1}\ and\ \ (e_1 \in DGT_1(e) \mid DGT_2(e) \mid DGT_3(e) or \mid DGT_4(e)) \tag{3.3}$$

GLT(e) : A set of Global Leader Table,

d (e, e_1) : The distance between entity e and entity e_1,

γ_{e_1} : An entity's Aura,

DGT$_i$(e_1) : A set of Group in the Quadrant, for i = 1~4.

3.2.3.1 Interchange protocol between entities

As mentioned before, the OT and GT should be interchanged periodically. The new method of table interchange between entities is listed as followed:

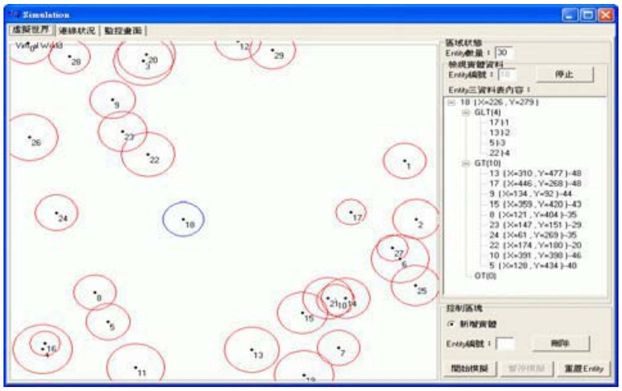
- If OT is not empty, the entities in the OT are choosing.
- If OT is empty but GT is not empty, the entities in the GT are choosing.
- Otherwise the entities in the GLT are choosing.

4. Simulation Result

The simulation system is a region in a network virtual world. There are two kinds of components in this environment such as server and entity components.

4.1. The network topology

In Fig. 9, it shows all of the entities' positions in the virtual world.



Entity數量：30

檢視實體資料

Entity編號：18

Entity三資料表內容：

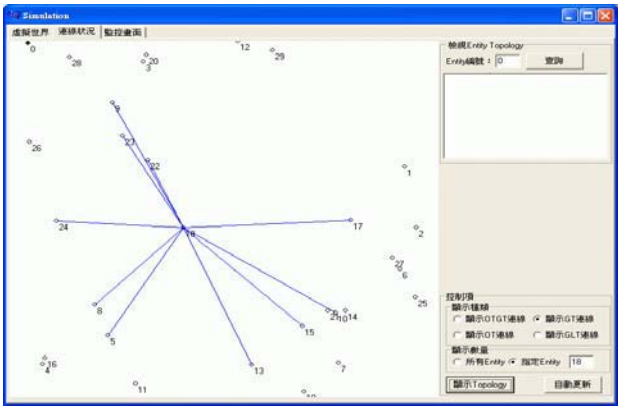
```
⊟ 18 (X=226 , Y=279 )
   ⊟ GLT(4)
        17 )-1
        13 )-2
        5 )-3
        22 )-4
   ⊟ GT(10)
        13 (X=310 , Y=477 )-48
        17 (X=446 , Y=268 )-48
        9 (X=134 , Y=92 )-44
        15 (X=359 , Y=420 )-43
        8 (X=121 , Y=404 )-35
        23 (X=147 , Y=151 )-29
        24 (X=61 , Y=269 )-35
        22 (X=174 , Y=180 )-20
        10 (X=391 , Y=398 )-46
        5 (X=128 , Y=434 )-40
     OT(0)
```

◉ 新增實體

Entity編號：

Fig. 9 The positions of all entities in the virtual world

In Fig. 10, it shows the connection of GT of the center entity. This entity only stores the near entities in its GT.



Entity編號：0

顯示連線

○ 顯示OTGT連線 ◉ 顯示GT連線

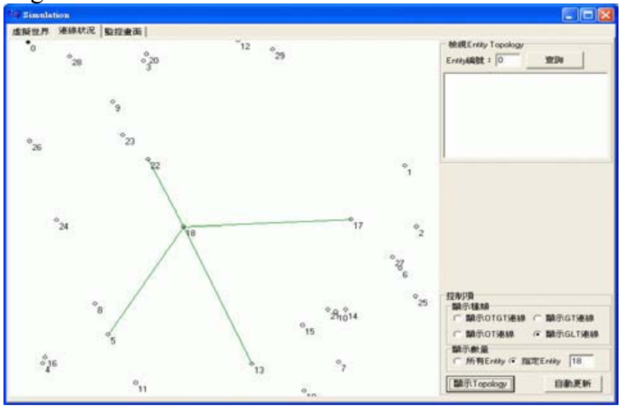
○ 顯示OT連線 ○ 顯示GLT連線

顯示數量

○ 所有Entity ◉ 指定Entity 18

Fig. 10 The connection of GT of the center entity

In Fig. 11, show the network topology of GLT. As the time goes by, it can also keep the Leader of each Group. It demonstrates the Group in Four Quadrants Algorithm.



Entity編號：0

顯示連線

○ 顯示OTGT連線 ○ 顯示GT連線

○ 顯示OT連線 ◉ 顯示GLT連線

顯示數量

○ 所有Entity ◉ 指定Entity 18

Fig. 11 The connection of Group Leader Table

In Fig. 12, it shows the network topology of virtual world. It demonstrates the connected graph of this virtual world.

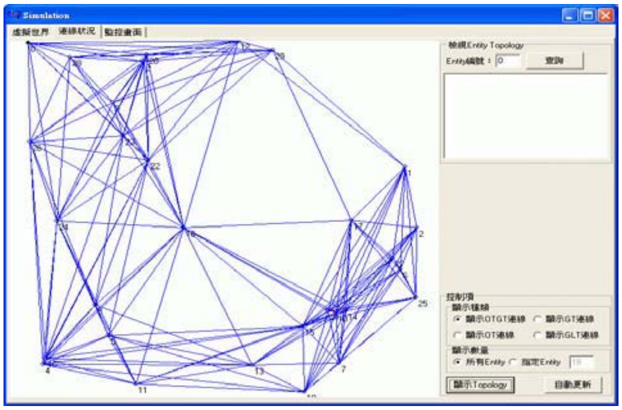
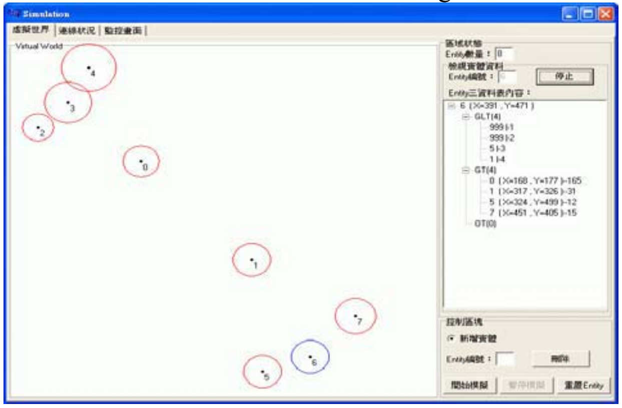


Fig. 12 The network topology of the virtual world

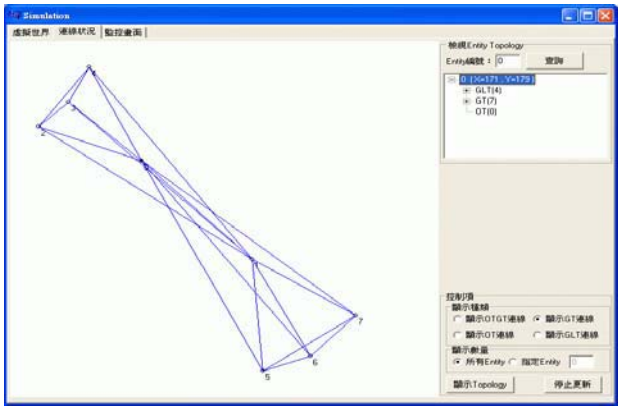
4.2. The recovery of some entities crash

In this section, when some entities are crashed or disconnected, the simulation results shows that how the environment is recovered as shown in Fig. 13 and 14.



```
6 (X=391 , Y=471 )
    GLT(4)
        999 )-1
        999 )-2
        5 )-3
        1 )-4
    GT(4)
        0 (X=168 , Y=177 )-165
        1 (X=317 , Y=326 )-31
        5 (X=324 , Y=499 )-12
        7 (X=451 , Y=405 )-15
    OT(0)
```

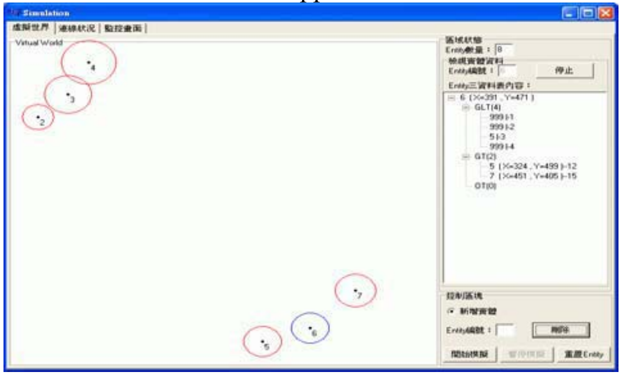
Fig. 13 the Virtual World



```
0 (X=171 , Y=179 )
    GLT(4)
    GT(7)
    OT(0)
```

Fig. 14 the network topology

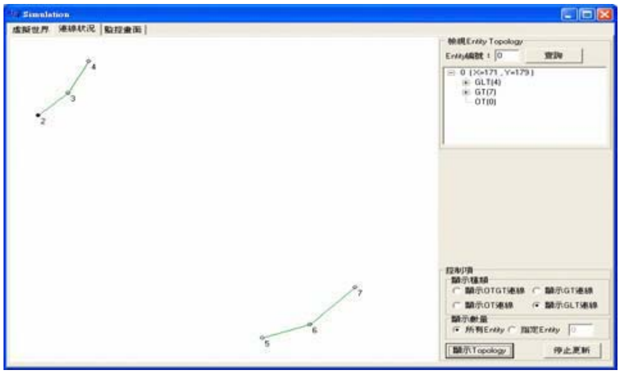
In Fig. 15, as compared with Fig. 13, two entities e0 and e1 are crashed and disappeared in the virtual world.



```
6 (X=391 , Y=471 )
    GLT(4)
        999 )-1
        999 )-2
        5 )-3
        999 )-4
    GT(2)
        5 (X=324 , Y=499 )-12
        7 (X=451 , Y=405 )-15
    OT(0)
```

Fig. 15 The topology of virtual world after two entities crashed

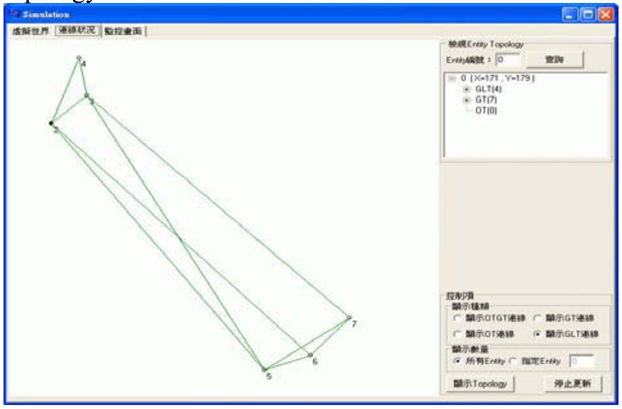
After a period of time, other entities cannot get information by interchange data and they became two connected graphs as shown in Fig. 16.



```
0 (X=171 , Y=179 )
    GLT(4)
    GT(7)
    OT(0)
```

Fig. 16 Two connected graphs because two entities crash

Final, as shown in Fig. 17, the new group leader is selection. The entities can get information of other entities' from the group leader and the network topology of virtual world is rebuilt.



```
0 (X=171 , Y=179 )
    GLT(4)
    GT(7)
    OT(0)
```

Fig. 17 The topology of virtual world after twice interchange information

5. Conclusion and Future Work

In this paper, a peer-to-peer transmission architecture is proposed to construct the virtual world. By this architecture every entity can deliver data to other entities without the help of the server. The transmission of the virtual world can still work when the server crashed. In this proposed architecture, an entity can send data to other entities by the Observable Table. In order to maintain a correct Observable Table, the data transmitted between its Observable Table and Global Table is referenced. The Group Leader Table is used to ensure the proposed topology is fully connected.

When an entity interchanges tables with other entities, it would get a lot of duplicated data. An algorithm should be designed to solve this problem in the future. In addition, how to extend this proposed peer-to-peer architecture to send data among in different region is also a future work.

6. References

[Chen 04] Jui-Fa Chen, Wei-Chuan Lin, Chih-Yu Jian, Heng-Yi Chiou, " A Searching Mechanism on a Distributed Virtual World: Six-Direction Simultaneous Search",Proceedings of the 10th IEEE International Symposium Pacific Rim Dependable Computing, Fast Abstracts, March 3-5, 2004, pp.. 13-14

[Grimaldi 99] R. P. Grimaldi, " An Introduction to Graph Theory", Discrete and Combinatorial Mathematics, Addison-Wesley Pub Co., ISBN: 0-201-30424-4, (1999), pp.477-546.

[Faisstnauer 00] C. Faisstnauer, D. Schmalstieg, W. Purgathofer, "Scheduling for very large virtual environments and networked games using visibility and priorities", Proceedings of the 2000 Distributed Simulation and Real-Time Applications (DS-RT 2000), 31-38 San Francisco, CA , USA, August 2000.

[Funkhouser 96] T. A. Funkhouser , "Network Topologies for Scalable Multi-User Virtual Environment.", Proceedings of the 1996 virtual Reality Annual International Symposium (VRAIS), 222-228, Santa Clara, Califormia, March 1996.

[Hawkinson 96] J. Hawkinson , "Guidelines for creation, selection, and registration of an Autonomous System (AS)", RFC1930, March 1996, http://www.rfc-editor.org

[Huang 03] Jiung-Yao Huang, Yi-chang Du, Chien-Min Wang, "Design of the server cluster to support avatar migration", Proceedings of 2003 IEEE Virtual Reality, 7-14, March 22-26, 2003.

[Keller 02] Keller, J. Simon, G., "Toward a peer-to-peer shared virtual reality", Proceedings of the 2002 Distributed Computing Systems Workshops, 695-700, Issy Moulineaux, France, July 2002.

[Macedonia 95] M. R. Macedonia, M. J. Zyda, D. P. Brutzman, and P. T. Barham, "Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments",Proc. 1995, IEEE Virtual Reality Annual International Symposium (VRAIS95), March 1995. pp. 11-15.

[Malkin 98] G. Malkin, "RIP Version 2", RFC2453, November 1998, http://www.rfc-editor.org

[Singhal 97] S.K. Singhal, B.Q. Nguyen, R. Redpath, J. Nguyen and M. Fraenkel, "InVerse: Designing an interactive universe architecture for scalability and extensibility", The Sixth IEEE International Symposium on High Performance Distributed Computing, 1997, pp.61-70.

[Singhal 99] S. Singhal and M. Zyda, "The Promises and Challenges of Networked Virtual Environments", Networked Virtual Environment – Design and Implementation, Addison-Wesley pub Co., ISBN: 0201325578, (1999), pp.8-16

[Wachowski 03] L. Wachowski and A. Wachowski. Matrix: Reloaded. Movie, 2003

[Malkin 98] G. Malkin, "RIP Version 2", RFC2453, November 1998, http://www.rfc-editor.org